Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 7192–7205
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# A high-order finite-volume algorithm for Fokker–Planck collisions in magnetized plasmas

Z. Xiong *, R.H. Cohen, T.D. Rognlien, X.Q. Xu

*Lawrence Livermore National Laboratory, Fusion Energy Program Livermore, CA 94550, USA*

Received 26 April 2007; received in revised form 18 March 2008; accepted 2 April 2008
Available online 15 April 2008

**Abstract**

A high-order finite-volume algorithm is developed for the Fokker–Planck Operator (FPO) describing Coulomb collisions in strongly magnetized plasmas. The algorithm uses a generic fourth-order reconstruction scheme on an unstructured grid in the velocity space spanned by parallel velocity and magnetic moment. By analytically mapping between different coordinates, it produces an accurate and density-conserving numerical FPO for an arbitrary choice of velocity space coordinates. A linearized FPO in constants-of-motion coordinates is implemented as an example of the present algorithm combined with a cut-cell merging procedure. Numerical tests include the thermalization of a test distribution with a background Maxwellian at a different temperature, and the return to isotropy for a distribution initialized with a velocity space loss-cone. Utilization of the method for a nonlinear FPO is straightforward but requires evaluation of the Trubnikov–Rosenbluth potentials.


*Keywords:* Fokker–Planck collisions; High-order scheme; Finite volume; Constants-of-motion coordinates

## 1. Introduction

The differential Fokker–Planck Operator (FPO) describes the particle collisions in a fully ionized plasma through shielded electrostatic Coulomb fields. Such Coulomb collisions are important in many systems, including laboratory plasma physics devices for basic studies, magnetic and inertial fusion, industrial material processing, and astrophysics. In magnetized plasma, because of the gyro-motion of the charged particles around magnetic field lines, the FPO is typically written in spherical coordinates spanned by $(v, \theta, \phi)$, where $v$ is the particle speed, $\theta$ is the pitch angle and $\phi$ the gyro-angle. For strong magnetic fields, many physical phenomena have characteristic time scales much longer than the gyro-period, and characteristic length scales much larger than the gyro-radius. In such cases, a gyro-averaging procedure may be applied and the resulting distribution function becomes independent of gyro-angle $\phi$. By further using a series expansion with Legendre polynomials $L_n(\cos\theta)$ in the $\theta$ direction, the evaluation of FPO is reduced to solving a series of one-dimen-

* Corresponding author. Present address: Schlumberger Sugar Land Technology Center. Tel.: +1 281 285 7331.
*E-mail address:* zxiong@gmail.com (Z. Xiong).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.04.004

sional equations of $v$ only [1,2]. Following this approach, Chang and Cooper [3] developed a finite difference scheme that conserves particle number density. This scheme was further extended by Epperlein [4] to conserve particle energy. Khabibrakhmanov and Khazanov [5] have recently solved these equations using a spectral collocation method.

The velocity coordinates $(v, \theta)$, though convenient for evaluating the FPO, are not particularly suitable for simulating spatially inhomogeneous plasmas where particle advection, including parallel streaming and perpendicular drifts, is important. To accurately compute the particle orbits, the velocity coordinates are often determined by the particular choice of the spatial advection schemes. In such cases, the FPO needs to be evaluated in the same coordinates and its discretization becomes truly two-dimensional. Chacon et al. [6] proposed a tensor formulation for two-dimensional FPO and studied the finite difference scheme in cylindrical coordinates for improved energy conservation. Other algorithms dealing with non-isotropic, multi-dimensional FPOs use Cartesian velocity coordinates directly [7,8]. Cartesian coordinates, however, are almost never used directly for strongly magnetized plasmas owing to the usefulness of averaging over the rapid particle gyromotion to remove one dimension from the computation. Recent attempts to couple the FPO with the Vlasov equation in different velocity coordinates have ignored the spatial dependence of the distribution function in the collision operator [9,10].

The so-called constants-of-motion coordinates, e.g. the total energy $E$ and the magnetic moment $\mu$, have been used in gyrokinetic simulations of fusion plasma with both particle [11] and continuum [12] formulations. For these simulations, it is critical that the passing and trapped particle orbits are accurately represented. The choice of $(\mu, E)$ coordinates is advantageous because $(\mu, E)$ remain constant along particle orbits (in the absence of collisions and time-varying fields), and the velocity coordinates $(\mu, E)$ are thus orthogonal to the spatial coordinates. For instance, the collisionless Vlasov equation is particularly simple when written in $(\mu, E)$ coordinates. To compute collisional effects accurately, the same constants-of-motion coordinates should be used in the FPO as well. The approach of using direct interpolation of the collision operator between different velocity coordinates has been found unsatisfactory, particularly with respect to the conservation properties. In this paper, we present an algorithm for computing the FPO in constants-of-motion coordinates based on a generic high-order finite volume scheme on unstructured grids, which is inherently particle-number-conserving. Although focusing on the constants-of-motion coordinates in this paper, our goal is to develop a numerical FPO that is accurate, conservative and easily applied to different coordinates systems.

The strategy is to first choose a convenient but fixed coordinate system, e.g. in this case $(v_\parallel, \mu)$, and then evaluate the FPO in these coordinates using a conservative, high-order finite volume scheme on a unstructured mesh. Here $v_\parallel$ is the velocity along the magnetic field. The finite volume discretization is inherently density conserving, and an unstructured mesh decouples the choice of coordinates and the gridding strategy. In this way, different velocity coordinates can be mapped *directly* onto the chosen $(v_\parallel, \mu)$ coordinates, with a regular grid in the former typically becoming an irregular and unstructured grid in the latter. After the mapping, the solution we obtain still maintains high-order accuracy and good conservation properties. In this sense, the evaluation of the FPO is independent of the choice of velocity coordinates, and the constants-of-motion coordinate set is but one such choice. For simplicity, the method is illustrated using a linearized FPO, where the collision diffusion coefficients are known by assuming the background particle distribution to be Maxwellian. For the nonlinear FPO, the diffusion coefficients need to be obtained first by solving Trubnikov–Rosenbluth potentials [13], which is an important but rather independent problem and shall be dealt with separately. Once the diffusion coefficients are known, the algorithm described here applies to the nonlinear FPO directly.

The remainder of the paper is organized as follows: the formulation of the FPO in $(v_\parallel, \mu)$ coordinates is given in Section 2 both for nonlinear and linearized cases. The high-order finite volume scheme on a general unstructured mesh is presented in Section 3. In Section 4, we describe the cut-cell method in $(\mu, E)$ space, and the choice of stencils for finite volume reconstruction. The numerical tests are presented in Section 5, and the concluding remarks are given in Section 6.

## 2. Fokker–Planck collision operator

Here the general Fokker–Planck collision operator is given in $(v_\parallel, \mu)$ coordinates, followed by the linearized version about a fixed Maxwellian distribution function describing the background field particles.

### 2.1. General form

Following Trubnikov [13], the Fokker–Planck collision operator $C$, written in a divergence form is,

$$\left.\frac{\partial f_\alpha}{\partial t}\right|_c = C(f_\alpha) = \sum_\beta C^{\alpha/\beta}(f_\alpha) = -\sum_\beta \frac{\partial S_i^{\alpha/\beta}}{\partial v_i}, \tag{1}$$

where $f_\alpha$ is the particle distribution function, and $S^{\alpha/\beta}$ is the flux of 'test' particles of species $\alpha$ due to collisions with 'field' particles of species $\beta$ (including $\beta = \alpha$). The flux $S^{\alpha/\beta}$ consists of a friction term and a diffusion term:

$$S_i^{\alpha/\beta} = \frac{F_i^{\alpha/\beta}}{m_\alpha} f_\alpha - D_{ik}^{\alpha/\beta} \frac{\partial f_\alpha}{\partial v_k}. \tag{2}$$

Using the so-called Trubnikov–Rosenbluth potentials [1,13], we can write

$$\frac{F_i^{\alpha/\beta}}{m_\alpha} = -\frac{m_\alpha}{m_\beta} n_\beta \Lambda^{\alpha/\beta} \frac{\partial h^\beta}{\partial v_i}, \quad D_{ik}^{\alpha/\beta} = -n_\beta \Lambda^{\alpha/\beta} \frac{\partial^2 g^\beta}{\partial v_i \partial v_j}, \tag{3}$$

where $\Lambda^{\alpha/\beta}$ is defined as

$$\Lambda^{\alpha/\beta} = \Lambda_c \left(\frac{4\pi Z_\alpha Z_\beta e^2}{m_\alpha}\right)^2, \tag{4}$$

and $\Lambda_c = \log(8\pi(n_e \lambda_D^3))$ is the Coulomb logarithm, $\lambda_D$ is the Debye length and $n_e$ is the electron density; $m$ is the particle mass, $n$ is the number density, $e$ is the fundamental unit charge, and $Z$ is the ion charge number.

The Trubnikov–Rosenbluth potentials are defined as

$$h^\beta = -\frac{1}{4\pi} \int \frac{f_\beta(\vec{v}')}{|\vec{v} - \vec{v}'|} \, d\vec{v}', \quad g^\beta = -\frac{1}{8\pi} \int |\vec{v} - \vec{v}'| f_\beta(\vec{v}') \, d\vec{v}', \tag{5}$$

where the distribution function is normalized such that its velocity integral is unity. Using the identities,

$$\nabla_v^2 |\vec{v} - \vec{v}'| = \frac{2}{|\vec{v} - \vec{v}'|}, \quad \nabla_v^2 \frac{1}{|\vec{v} - \vec{v}'|} = -4\pi\delta(\vec{v} - \vec{v}'), \tag{6}$$

it is easily established that

$$\nabla_v^2 g^\beta = h^\beta, \quad \nabla_v^2 h^\beta = f_\beta. \tag{7}$$

### 2.2. $v_\parallel - \mu$ coordinates

For magnetized plasmas, the parallel velocity $v_\parallel$ is along the direction of the local magnetic field line and the magnetic moment is defined as $\mu = \frac{m_\alpha}{2B} v_\perp^2$, where $B$ is the local magnetic field strength and $v_\perp$ is the perpendicular velocity. Note that $v_\parallel$ and $\mu$ are a convenient choice for velocity space coordinates for magnetized plasmas after gyro-averaging, because $\mu$ is a constant of motion, and $v_\parallel$, unlike $E$, makes $f$ a single-valued function in velocity space. The Jacobian of the $(v_\parallel, \mu)$ coordinates is a constant. Moreover, the conservative form of the Vlasov equation can be simply expressed in $(v_\parallel, \mu)$ space.

Ignoring the gyro-angle dependence, the resulting axisymmetric FPO can be written in $(v_\parallel, \mu)$ coordinates as,

$$C(f_\alpha) = \frac{\partial \Gamma_{v_\parallel}}{\partial v_\parallel} + \frac{\partial \Gamma_\mu}{\partial \mu}, \tag{8}$$

where the fluxes $\Gamma_{v_\parallel}$ and $\Gamma_\mu$ are defined as

$$\Gamma_{v_\parallel} = D_v f_\alpha + D_{vv} \frac{\partial f_\alpha}{\partial v_\parallel} + D_{v\mu} \frac{\partial f_\alpha}{\partial \mu}, \tag{9}$$

$$\Gamma_\mu = D_\mu f_\alpha + D_{\mu v} \frac{\partial f_\alpha}{\partial v_\parallel} + D_{\mu\mu} \frac{\partial f_\alpha}{\partial \mu}, \tag{10}$$

and the coefficients are given by

$$D_v = n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha}{m_\beta} \frac{\partial h^\beta}{\partial v_\parallel}, \tag{11}$$

$$D_{vv} = -n_\beta \Lambda^{\alpha/\beta} \frac{\partial^2 g^\beta}{\partial v_\parallel^2}, \tag{12}$$

$$D_{v\mu} = -2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha}{B} \mu \frac{\partial^2 g^\beta}{\partial v_\parallel \partial \mu}, \tag{13}$$

$$D_\mu = 2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha^2}{B m_\beta} \mu \frac{\partial h^\beta}{\partial \mu}, \tag{14}$$

$$D_{\mu v} = -2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha}{B} \mu \frac{\partial^2 g^\beta}{\partial v_\parallel \partial \mu}, \tag{15}$$

$$D_{\mu\mu} = -2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha^2}{B^2} \mu \left( 2\mu \frac{\partial^2 g^\beta}{\partial \mu^2} + \frac{\partial g^\beta}{\partial \mu} \right). \tag{16}$$

The Trubnikov–Rosenbluth potentials satisfy

$$\frac{\partial^2 g^\beta}{\partial v_\parallel^2} + \frac{m_\alpha}{B} \frac{\partial}{\partial \mu} \left( 2\mu \frac{\partial g^\beta}{\partial \mu} \right) = h^\beta, \tag{17}$$

$$\frac{\partial^2 h^\beta}{\partial v_\parallel^2} + \frac{m_\alpha}{B} \frac{\partial}{\partial \mu} \left( 2\mu \frac{\partial h^\beta}{\partial \mu} \right) = f_\beta. \tag{18}$$

Eqs. (8)–(18) constitute the general nonlinear FPO in $(v_\parallel, \mu)$ coordinates.

### 2.3. Linearization

The nonlinear FPO (8)–(18) may be linearized if the collisions between test particles and background particles are more important than collisions among the test particle themselves. Moreover, this approximation is useful if the background plasma is maintained close to a Maxwellian, i.e.

$$f_\beta(v) = F_{\mathrm{M}}^\beta(v) = (\sqrt{\pi} v_{t\beta})^{-3} \exp(-v^2/v_{t\beta}^2), \tag{19}$$

where $v_{t\beta} = (2T_\beta/m_\beta)^{\frac{1}{2}}$ is the thermal velocity of field particles, and $T_\beta$ is its mean temperature.

In such cases, the Trubnikov–Rosenbluth potentials depend only on the particle speed $v$, i.e. $h^\beta = h^\beta(v)$ and $g^\beta = g^\beta(v)$, where

$$v = \sqrt{v_\parallel^2 + 2\mu B/m_\alpha}. \tag{20}$$

As a result, the derivatives of $h$ and $g$ are

$$\frac{\mathrm{d}h^\beta}{\mathrm{d}v} = \frac{1}{4\pi v^2} H\left( \frac{v}{v_{t\beta}} \right), \tag{21}$$

$$\frac{\mathrm{d}g^\beta}{\mathrm{d}v} = \frac{T_\beta}{8\pi m_\beta v^2} H\left( \frac{v}{v_{t\beta}} \right) - \frac{1}{8\pi} G\left( \frac{v}{v_{t\beta}} \right), \tag{22}$$

$$\frac{\mathrm{d}^2 g^\beta}{\mathrm{d}v^2} = -\frac{T_\beta}{4\pi m_\beta v^3} H\left( \frac{v}{v_{t\beta}} \right), \tag{23}$$

where the error function $G(x)$, and function $H(x)$ are defined as

$$G(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-y^2} \mathrm{d}y, \quad H(x) = \frac{4}{\sqrt{\pi}} \int_0^x y^2 \mathrm{e}^{-y^2} \mathrm{d}y = G(x) - xG'(x). \tag{24}$$

The coefficients in (9) and (10) are now given by

$$\widetilde{D}_v = n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha}{m_\beta} \frac{v_\parallel}{v} \frac{\mathrm{d}h^\beta}{\mathrm{d}v}, \tag{25}$$

$$\widetilde{D}_{vv} = -n_\beta \Lambda^{\alpha/\beta} \left( \frac{2\mu B}{m_\alpha v^3} \frac{\mathrm{d}g^\beta}{\mathrm{d}v} + \frac{v_\parallel^2}{v^2} \frac{\mathrm{d}^2 g^\beta}{\mathrm{d}v^2} \right), \tag{26}$$

$$\widetilde{D}_{v\mu} = 2n_\beta \Lambda^{\alpha/\beta} \left( \frac{\mu v_\parallel}{v^3} \frac{\mathrm{d}g^\beta}{\mathrm{d}v} - \frac{\mu v_\parallel}{v^2} \frac{\mathrm{d}^2 g^\beta}{\mathrm{d}v^2} \right), \tag{27}$$

$$\widetilde{D}_{\mu} = 2n_\beta \Lambda^{\alpha/\beta} \frac{m_\alpha \mu}{m_\beta v} \frac{\mathrm{d}h^\beta}{\mathrm{d}v}, \tag{28}$$

$$\widetilde{D}_{\mu v} = 2n_\beta \Lambda^{\alpha/\beta} \left( \frac{\mu v_\parallel}{v^3} \frac{\mathrm{d}g^\beta}{\mathrm{d}v} - \frac{\mu v_\parallel}{v^2} \frac{\mathrm{d}^2 g^\beta}{\mathrm{d}v^2} \right), \tag{29}$$

$$\widetilde{D}_{\mu\mu} = 2n_\beta \Lambda^{\alpha/\beta} \left[ \left( \frac{2\mu^2}{v^3} - \frac{m_\alpha \mu}{vB} \right) \frac{\mathrm{d}g^\beta}{\mathrm{d}v} - \frac{2\mu^2}{v^2} \frac{\mathrm{d}^2 g^\beta}{\mathrm{d}v^2} \right]. \tag{30}$$

Eqs. (8)–(10) with relations (25)–(30) constitute the linearized FPO in $(v_\parallel, \mu)$ coordinates.

## 3. Finite volume discretization

### 3.1. General formula

To solve the FPO numerically, consider a fixed partition of the $(v_\parallel, \mu)$ plane by a set of non-overlapping polygonal sub-domains $\Omega_i$, $i = 1, \ldots, N$, each constituting a cell. Within each cell $\Omega_i$, the Fokker–Planck equation

$$\left.\frac{\partial f}{\partial t}\right|_c = \nabla \cdot \Gamma = \frac{\partial \Gamma_{v_\parallel}}{\partial v_\parallel} + \frac{\partial \Gamma_\mu}{\partial \mu} \tag{31}$$

can be integrated using Gauss' theorem to obtain the basic form of the finite-volume discretization,

$$\left.\frac{\partial \overline{f_i}}{\partial t}\right|_c = \frac{1}{V_i} \sum_{j \in \partial\Omega_i} \int (\Gamma \cdot n_j)\, \mathrm{d}S_j \quad \text{where } \bar{f}_i = \frac{1}{V_i} \int_{\Omega_i} f\, \mathrm{d}v. \tag{32}$$

Here $\bar{f}_i$ is the cell-averaged value of distribution function $f$ on cell $i$, with $V_i$ and $\partial\Omega_i$ being its volume and boundary. The $\bar{f}_i$ are the fundamental variables solved in the finite volume method, and their rates of change are determined by the fluxes across the cell edges. To evaluate the fluxes $\Gamma_{v_\parallel}$ and $\Gamma_\mu$ in (9) and (10), we need the point-wise values of $f$ and its derivatives on the cell edges. Thus a central piece of the finite-volume algorithm is the so-called reconstruction, that is, determining the point-wise values of $f(v_\parallel, \mu)$ from the known cell-averaged values $\bar{f}_i$.

The reconstruction algorithm uses the cell-averaged function value on a given cell, and those of the neighboring cells, to form a local approximation to $f(v_\parallel, \mu)$ in such a way that when this approximate function is averaged over any cell involved, the known cell-averaged value is recovered. Hence let us assume $f$ to be a generic function of $v_\parallel$ and $\mu$, and we can approximate $f$ in the neighborhood of a cell $p$ by a polynomial of $v_\parallel$ and $\mu$, i.e.,

$$f_p(v_\parallel, \mu) \approx \sum_{i=0}^{n} \sum_{j=0}^{n-i} a_{ij} (v_\parallel - v_{\parallel o})^i (\mu - \mu_o)^j. \tag{33}$$

Here $n$ is the desired order of the approximating polynomial, and $a_{ij}$ are the unknown coefficients. For a given order $n$, the total number of the coefficients $a_{ij}$ is $(n+1)(n+2)/2$. The set $(v_{\parallel o}, \mu_o)$ is an appropriately chosen reference point. Here we observe that if $f_p$ is a unique approximation based on $(v_{\parallel o}, \mu_o)$, i.e., given all the $a_{ij}$ determined uniquely, then it can be equivalently written for another reference point $(v'_{\parallel o}, \mu'_o)$. The new coeffi-

cients will become $a'_{ij} = a'_{ij}(a_{mn}, v_{\|o}, v'_{\|o}, \mu_o, \mu'_o)$, but $f_p$ itself remains the same, i.e., the choice of $(v_{\|o}, \mu_o)$ is not essential to the approximation. Consequently, we select the origin $v_{\|o} = \mu_o = 0$ as a convenient choice and is used throughout the paper. The advantage of a unique origin will be seen in the next section.

### 3.2. High-order reconstruction

To obtain an accurate representation of the FPO with modest grid resolution, we choose a fourth-order approximation, $n = 4$. Therefore 15 coefficients $a_{ij}$ need to be determined in (33). To simplify the notation, we reorder the $a_{ij}$s by a subscript $k$, where $k = 0, \ldots, 14$, and for each $k$, the corresponding exponents of $v_\|$ and $\mu$ are denoted as $i_k$ and $j_k$, where $i_k + j_k \leqslant n$. So each $a_k$ is the coefficient of a base polynomial of order $i_k + j_k$. To determine $a_k$ on a particular cell, we first construct a stencil consisting of cell $p$ and its 14 neighboring cells. The construction of such a stencil is discussed in detail in Section 4.2. Once the stencil is chosen, integrating (33) over each cell of the stencil yields a system of linear equations for the $a_k$:

$$B_{mk}\, a_k = \bar{f}_m, \quad m, k = 0, 1, \ldots, 14, \tag{34}$$

where the matrix $B_{mk} = [b_{mk}]$, and $b_{mk}$ is given by

$$b_{mk} = \frac{1}{V_m} \int_{\Omega_m} v_\|^{i_k} \mu^{j_k}\, \mathrm{d}v_\|\, \mathrm{d}\mu, \quad \text{where } V_m = \int_{\Omega_m} \mathrm{d}v_\|\, \mathrm{d}\mu. \tag{35}$$

Here $b_{mk}$ is the cell averaged value of the $k$th base polynomial on the $m$th cell. Note that the $b_{mk}$ depends only on the metrics of the $m$th cell but not on $\bar{f}_m$. More importantly, the value of $b_{mk}$ is independent of the specific stencil used. This results from the fact that all the polynomials in $b_{mk}$ are based on a unique global origin (see discussion in Section 3). Thus each $b_{mk}$ needs to be computed only once.

To evaluate $b_{mk}$ on a cell, we use the simplex of a 2D plane, a triangle, because any polygonal cell can be divided into one or several triangles (in 3D it is a tetrahedron). It suffices to integrate the base polynomials over a simplex. Let $\Delta = \{(v_{\|0}, \mu_0), (v_{\|1}, \mu_1), (v_{\|2}, \mu_2)\}$ be such a simplex; then for fourth-order accuracy, we use the 16-point Gaussian quadrature

$$\int_\triangle f(v_\|, \mu)\, \mathrm{d}v_\|\, \mathrm{d}\mu = \sum_{i=0}^{15} w_i f(v_{\|i}, \mu_i) = J \sum_{i=0}^{15} w_i f(\widetilde{v}_{\|i}, \widetilde{\mu}_i), \tag{36}$$

where the $(\tilde{v}_\|, \tilde{\mu})$ are the Gaussian abscissa in the standard triangle at (0, 0), (0, 1) and (1, 0). The transformation between $(v_{\|i}, \mu_i)$ and $(\widetilde{v}_{\|i}, \widetilde{\mu}_i)$ is given by

$$v_{\|i} = v_{\|0} + (v_{\|1} - v_{\|0})\widetilde{v}_{\|i} + (v_{\|2} - v_{\|0})\widetilde{\mu}_i \tag{37}$$

$$\mu_i = \mu_0 + (\mu_1 - \mu_0)\widetilde{v}_{\|i} + (\mu_2 - \mu_0)\widetilde{\mu}_i. \tag{38}$$

Here $J = |(v_{\|1} - v_{\|0})(\mu_2 - \mu_0) - (v_{\|2} - v_{\|0})(\mu_1 - \mu_0)|$ is the Jacobian. The numerical values of the Gaussian weights $w_i$ and abscissa $(\widetilde{v}_{\|i}, \widetilde{\mu}_i)$ are given in the Appendix. Once the $b_{mk}$ are known, the coefficients $a_k$ can be obtained by solving (34), e.g. using the Gaussian elimination method. Substituting the coefficients into (33), we obtain the pointwise reconstruction formula for $f(v_\|, \mu)$. In addition, the pointwise derivatives $f_{v_\|}$ and $f_\mu$, needed for flux evaluations, follow directly,

$$f_{v_\|} = \sum_{i_k=1}^{n} \sum_{j_k=0}^{n-i_k} i_k a_k v_\|^{i_k - 1} \mu^{j_k}, \quad f_\mu = \sum_{i_k=0}^{n} \sum_{j_k=1}^{n-i_k} j_k a_k v_\|^{i_k} \mu^{j_k - 1}. \tag{39}$$

To integrate the fluxes across a cell edge, we use fourth-order Gaussian quadrature. For example, given the fluxes $\Gamma_p = \vec{\Gamma} \cdot \vec{n}$ at three Gaussian points on an edge, the flux across the edge is obtained by (after mapping the edge into domain $[-1, 1]$)

$$\int_l \Gamma\, \mathrm{d}l = \sum_{p=0}^{2} c_p \Gamma_p = \frac{l}{2} \sum_{p=0}^{2} c_p \Gamma(\tilde{x}_p), \tag{40}$$

here $l$ is the length of the edge, $c_p = \{5/9, 8/9, 5/9\}$ are the weights, and $\tilde{x}_p = \{-\sqrt{15}/5, 0, \sqrt{15}/5\}$ are the Gaussian abscissa on $[-1, 1]$.

Since each interior cell edge belongs to two neighboring cells, the flux across it can be computed from either cell. To have a conservative scheme, a unique definition of the flux must be used. Here we use an upwind approach, that is, assuming edge $pq$ is shared by cell $p$ and $q$, the flux $F$ is chosen according to

$$\Gamma = \begin{cases} \int \Gamma_{pq} \cdot n_{pq} \, \mathrm{d}l & \text{if } \int \Gamma_{pq} \cdot n_{pq} \, \mathrm{d}l > 0, \\ \int \Gamma_{qp} \cdot n_{qp} \, \mathrm{d}l & \text{otherwise.} \end{cases} \tag{41}$$

Here $\Gamma_{pq}$ and $\Gamma_{qp}$ are the computed point-wise fluxes from cell $p$ to cell $q$ and vice versa, and $n_{pq}$ is the unit normal of edge $pq$ pointing from $p$ to $q$.

We note a few features of this reconstruction method: (1) unlike in the conventional finite-volume method, the current reconstruction need not specify the location of $\bar{f}_i$, which is only associated with a cell. (2) The metric information of each cell is included automatically in the base polynomial through $b_{mk}$. So the reconstruction applies equally to structured or unstructured meshes with arbitrary cell shapes, and (3) once the $b_{mk}$ are computed initially for each cell, forming the matrix $B_{mk}$ becomes very simple because all its elements $(b_{mk})$ are known.

A properly chosen stencil is important to the reconstruction algorithm. The minimum requirement is that the resulting matrix $B_{mk}$ should be invertible and well-conditioned. In practice, the stencil for a particular cell should form a simply-connected domain in the neighborhood of this cell and have sufficient support in both directions to avoid a singularity in the resulting matrix. Since the FPO is a convection–diffusion operator in velocity space, we may also take into account the direction of the edge fluxes in forming the stencil. This will be discussed further in the context of constants-of-motion coordinates.

## 4. Constants-of-motion coordinates

Since the reconstruction scheme does not require the mesh to be regular or structured, the present algorithm can be applied to the case where the $(v_\parallel, \mu)$ mesh is mapped from another mesh in a different velocity coordinate system.

One choice of velocity space coordinates for the gyrokinetic plasma simulations, as noted in Section 1, is the total energy $E$ and the magnetic moment $\mu$. In the absence of collisions, $E$ and $\mu$ are conserved along particle orbits in the equilibrium (time-independent) fields. So for the Vlasov equation, the numerical approximation of the spatial derivative is effectively decoupled from the velocity space operation. This prevents the spatial approximation from introducing numerical diffusion into velocity space and is a particularly good choice for accurately calculating particle orbits [12]. An example of a regular $(\mu, E)$ mesh is shown in Fig. 1a, where the boundary $v_\parallel = 0$ is a straight line cutting through the grid and separating the physical zone (above, $v_\parallel^2 > 0$) from a non-physical zone (below, $v_\parallel^2 < 0$).

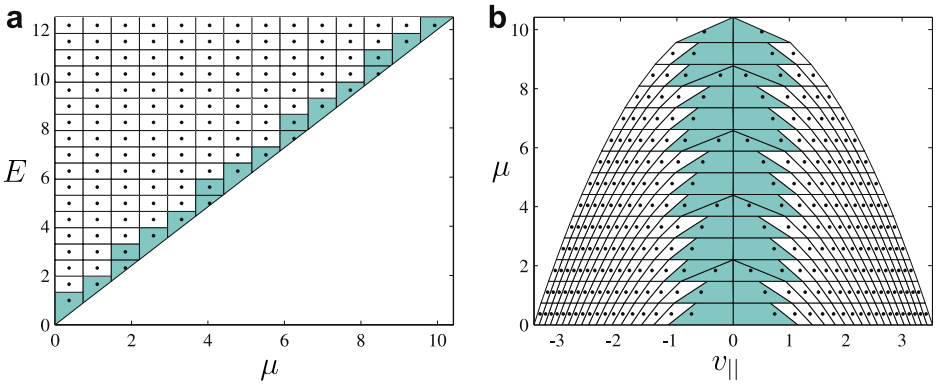


Fig. 1. Velocity space represented in $(\mu, E)$ and the corresponding $(v_\parallel, \mu)$ coordinates.

Existing FPOs typically are not written in constants-of-motion coordinates, so to compute particle collisions, we adopt the following strategy. First, we employ a cell cutting and merging technique (described below) to treat the cells being cut by the $v_\parallel = 0$ boundary. This procedure avoids arbitrarily small cut cells that can limit the time step size by the CFL constraint. Then, the resulting $(\mu, E)$ grid is mapped into $(v_\parallel, \mu)$ space, and the finite volume algorithm described in Section 3 is applied. The resulting cut cell may have various shapes and different connectivity patterns. For instance, Fig. 1b shows the corresponding $(v_\parallel, \mu)$ mesh mapped directly from the $(\mu, E)$ mesh in Fig. 1a. Note that the $v_\parallel = 0$ boundary now becomes the vertical axis about which the mesh is symmetric. The left- and right-half planes correspond to particles with different signs of the parallel velocity.

### 4.1. Cell merging and mapping

The cell cutting and merging procedure starts by computing the cell center of the regular background $(\mu, E)$ grid. If the center of a cell is above the $v_\parallel = 0$ boundary (physical domain), this cell will be retained. Otherwise, the cell will be discarded. If part of the cell that is discarded is in the physical zone, this part is merged into its neighboring cells. Following Ye et al. [14], we choose to merge cells in the $E$ direction. Since the $v_\parallel = 0$ boundary is a straight line, there are a total of four different types of cut cells as shown in Fig. 2. Note that depending on the slope of the $v_\parallel = 0$ boundary, i.e., the local $B$ field, there could be no neighboring cells available in the $E$ direction at the upper-right corner. In such cases, we merge the cut cells in $\mu$ direction. So the last such cell in Fig. 2 can appear only at the upper right corner with highest values of $E$ and $\mu$. As a result of merging, some of the cut-cell edges may be shared by more than one neighboring cell (see Fig. 1a). When summing the edge fluxes for this cell, contributions from different neighboring cells must be accounted for appropriately.

The mapping between $(\mu, E)$ and $(v_\parallel, \mu)$ coordinates is

$$v_\parallel = \pm\sqrt{E - \mu B - q\Phi}, \tag{42}$$

where $B$ is the magnetic field, and $\Phi$ is the electric potential. The variable $\mu$ itself is unchanged but now becomes the second coordinate in $(v_\parallel, \mu)$. Through this mapping, each $(\mu, E)$ cell (cut or regular cell) will be mapped into two symmetric $(v_\parallel, \mu)$ cells with opposite signs of $v_\parallel$. Because the mapping is analytic, particle conservation holds in both coordinates. In (42), a straight cell boundary of constant $E$ is not mapped strictly into a straight line boundary in $(v_\parallel, \mu)$; this poses no problem, as we only need to map the vertices of a cell. The initialization of $\bar{f}$, as well as the ensuing computations, are carried out in $(v_\parallel, \mu)$ space. This procedure is equivalent to changing the cell boundary in $(\mu, E)$ to a slightly curved line such that when mapped to $(v_\parallel, \mu)$, it becomes straight line. Note that the change of the cell shape in $(\mu, E)$ only negligibly affects the advection scheme in the equilibrium fields of physical space, because each particle orbit in physical space, is represented by a fixed point in $(\mu, E)$ space. Thus, the same collisional effects on the particle's orbit can be computed by FPOs using slightly different cells surrounding the same fixed point. One might also imagine applying the finite volume method directly in $(\mu, E)$ coordinates. However near the turning point boundary $v_\parallel = 0$, it is difficult for the reconstruction to include cells on the other sheet. With $(v_\parallel, \mu)$ coordinates, there is no multi-sheet problem, and the reconstruction at the turning point boundary is the same as anywhere else.

### 4.2. Reconstruction stencil

As mentioned earlier, the fourth-order reconstruction scheme requires a 15-cell stencil. Since the FPO represents convection and diffusion in velocity space, and the direction of the edge fluxes is not known *a priori*, we choose the stencil nearly centered on the cell on which the reconstruction is sought.

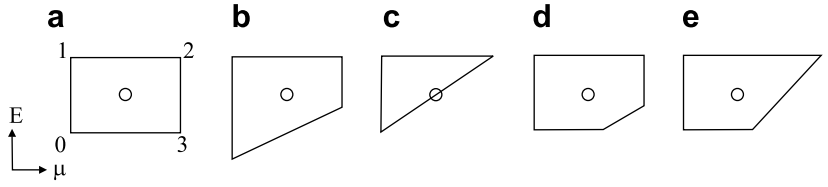


Fig. 2. Computational cells in $(\mu, E)$ coordinates. (a) is a regular cell and (b–e) are cut cells at $v_\parallel = 0$ boundary. (b) and (e) have been merged with neighboring cells. (e) only appears at the corner of highest $E$ and $\mu$.

Since each cell in $(\mu, E)$ corresponds to two cells in $(v_\parallel, \mu)$, to make the notation clear, we may describe the stencil using $(\mu, E)$ cells that are Cartesian except near the $v_\parallel = 0$ boundary.

Each cell, regular or cut (with merging), has one and only one cell center. These centers are the cell centers of the background grid; for the cut cells, they are typically not the geometric centers. Nevertheless, the cell center index serves as a unique label for each cell. The cell where the reconstruction is sought is given index $(i, j)$. Then corresponding to Fig. 3, besides cell $(i, j)$, the base stencil consists of 14 cells with the index shift shown in Table 1.

In Fig. 3, such a stencil is shown in both coordinates where all the cells in this stencil belong to the same $v_\parallel$ sheet in $(\mu, E)$. The numbers marked on the cells in $(\mu, E)$ and $(v_\parallel, \mu)$ coordinates show the mapping. In Fig. 4, a stencil with its center cell close to the turning point boundary is shown. Near the turning point boundary, some cells in the stencil are not available because the index shift places them in the non-physical zone $(v_\parallel^2 < 0)$. This means that the relevant cells are on the other sheet, i.e., the adjacent plane with the opposite sign of $v_\parallel$ axis in $(v_\parallel, \mu)$ space. The numbers in the parentheses designate such cells on the other sheet with the same $(\mu, E)$ indices that need to be included in the stencil.

In general, the stencil need not be static; its composition can change with time according to the direction of the local flux. Examples in one-dimensional systems include upwind, TVD, and ENO/WENO schemes.

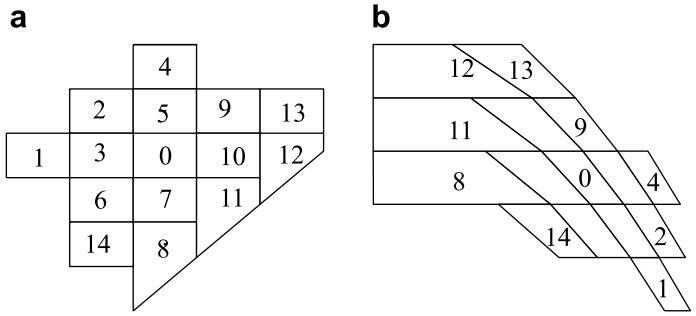


Fig. 3. A 15-cell stencil in (a) the $(\mu, E)$ coordinates and (b) the corresponding $(v_\parallel, \mu)$ coordinates. This stencil does not cross the cut-cell boundary, and only part of the corresponding cells in $(v_\parallel, \mu)$ coordinates are marked.

Table 1
Index shifts relative to cell $(i, j)$ in a 15-cell stencil

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\Delta i_k$ | −2 | −1 | −1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | −1 |
| $\Delta j_k$ | 0 | 1 | 0 | 2 | 1 | −1 | −1 | −2 | 1 | 0 | −1 | 0 | 1 | −2 |

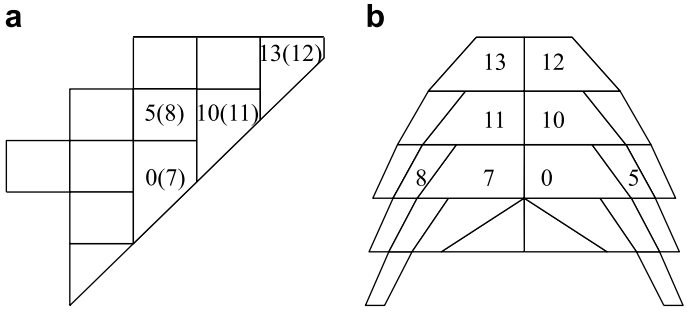


Fig. 4. A 15-cell stencil in the (a) $(\mu, E)$ and (b) corresponding $(v_\parallel, \mu)$ coordinates. This stencil crosses cut-cell boundary, and the overlapping cells are marked in both coordinates.

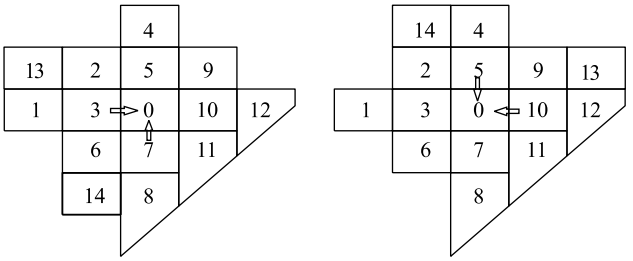


Fig. 5. Dynamic upwind stencil consisting of 15 cells in $(\mu, E)$ coordinates. The arrows indicate the direction of the net edge fluxes of cell 0.

Although for a multi-dimensional system a complete theory is still lacking, in practice there are ways to incorporate a dynamic stencil. For instance, a slight variation of the above static stencil can be made to form a stencil with a certain degree of upwinding. The idea is to keep the maximum extension of the stencil in $(\mu, E)$ directions the same, but choose the last two cells, 13 and 14, based on the direction of the edge fluxes of cell 0. Hence, the index shift of the cell 13 and 14 can be determined by

$$\Delta i_{13} = \begin{cases} -2 & \text{if } \Gamma_E > 0 \\ 2 & \text{otherwise,} \end{cases} \quad \text{and} \quad \Delta j_{14} = \begin{cases} -2 & \text{if } \Gamma_\mu > 0 \\ 2 & \text{otherwise.} \end{cases} \tag{43}$$

Here $\Gamma_E$ and $\Gamma_\mu$ are the net fluxes through the edge of the cell 0, which are obtained at the previous time step. In Fig. 5, two examples are shown for the location of cells 13 and 14 based on the sign of $\Gamma_E$ and $\Gamma_\mu$. The dynamic stencil can adjust itself to be biased towards the upwind direction of the edge fluxes, such that it offers better numerical stability and robustness compared to a static stencil.

To use the stencil near a boundary, ghost cells are needed. At the top boundary $E = E_{\max}$, we add two more rows of ghost cells and specify their values by assuming the distribution function decays exponentially in the $E$ direction. Similarly, at $\mu = 0$ boundary, we add two more columns of cells and assign their cell-averaged values by linear extrapolation based on three interior neighboring cells.

After merging the cut-cells and mapping the stencils, the FPO in constant-of-motion coordinates can now be evaluated directly in the $(v_\parallel, \mu)$ coordinates.

## 5. Numerical tests

### 5.1. Reconstruction accuracy

To test the accuracy of the reconstruction scheme, we choose a drifting Maxwellian with normalized unit density and mean temperature. The normalized drift velocity is $v_m = 0.1$. The upper grid boundaries of velocity space are set at $E_{\max} = \mu_{\max} = 16$, and the magnetic field is $B = 1.2$. Here for the cells close to $\mu = 0$, a biased stencil made of only interior cells is used, and the flux crossing the $\mu = 0$ boundary is set to zero. Given the initial cell-averaged distribution function $\bar{f}$, we compute the values of $f$ at cell corners, the mean drift velocity and the mean temperature using the reconstruction scheme. The last two correspond to the first and second moments of the distribution function $f$. The averaged pointwise reconstruction errors at different grid resolutions are summarized in Table 2. The error of the function values at the cell corners confirms the fourth-order accuracy of the reconstruction scheme. The errors of the mean velocity and temperature decrease even faster than that of the distribution function itself due to the integral effects in moment calculation, but the convergence rate is less consistent. Moreover, the accuracy of the reconstruction scheme is also important for determining the numerical conservation of the particle's energy. Unlike density, the particle's energy is not perfectly conserved in finite volume methods. To test numerical energy conservation, a nonlinear self-collision of a drifting Maxwellian was computed using the current FPO with the Trubnikov–Rosenbluth potentials supplied by the collision package CQL. It is found that the change of the mean energy after three collision times is less than 0.2% [15]. So the fourth-order reconstruction scheme also enables accurate numerical energy conservation for practical purposes.

Table 2
Mean pointwise errors of reconstruction for a normalized drift Maxwellian

| $N_\mu \times N_E$ | $\frac{1}{N}\sum_{i=0}^{N}\lvert f_i - f_M\rvert$ | $\frac{1}{N}\lvert U - U_M\rvert$ | $\frac{1}{N}\lvert T - T_M\rvert$ |
|---|---|---|---|
| $20 \times 20$ | $5.8594 \times 10^{-4}$ | $3.5225 \times 10^{-6}$ | $3.6900 \times 10^{-5}$ |
| $40 \times 40$ | $3.7685 \times 10^{-5}$ | $1.8083 \times 10^{-7}$ | $5.6541 \times 10^{-7}$ |
| $80 \times 80$ | $2.5080 \times 10^{-6}$ | $5.5365 \times 10^{-9}$ | $6.9256 \times 10^{-9}$ |
| $160 \times 160$ | $1.7617 \times 10^{-7}$ | $1.6416 \times 10^{-10}$ | $2.8535 \times 10^{-10}$ |

The columns are grid resolution, values at the cell corners, the mean drift velocity $U$, and the mean temperature $T$. Upper velocity space boundaries are at $E_{max} = \mu_{max} = 16$ and the magnetic field $B = 1.2$.

### 5.2. Thermalization of a test distribution function

We test the linear FP collision operator on the thermalization problem, which describes the thermal relaxation rate of a computed species $\alpha$ on a background fixed Maxwellian species $\beta$ [13]. Assuming the distribution of $\alpha$ particles also remains Maxwellian during the relaxation, the theoretical rate of temperature change for species $\alpha$ is given by

$$\frac{dT_\alpha}{dt} = -\frac{8m_\alpha}{3m_\beta\sqrt{\pi}}\frac{T_\alpha - T_\beta}{\tau^{\alpha/\beta}(T_\alpha + \frac{m_\alpha}{m_\beta}T_\beta)}. \tag{44}$$

Here $\tau^{\alpha/\beta}$ is the basic relaxation time defined by

$$\tau^{\alpha/\beta}(\epsilon) = \frac{\sqrt{m_\alpha}}{\pi\sqrt{2}e_\alpha^2 e_\beta^2}\frac{\epsilon^{3/2}}{\Lambda_c n_\beta}. \tag{45}$$

The time dependent FPO equation in $(\mu, E)$ space is solved using a standard fourth-order Runge–Kutta scheme. The background temperature can be either higher (heating) or lower (cooling) than the initial test particle temperature. The physical parameters are chosen as follows: $m_\alpha = m_\beta = 2m_p$, $e_\alpha = e_\beta = e$, where $m_p$ is the proton mass, and $e$ the magnitude of the electron charge. The density of the background distribution is $n_\beta = 10^{14}\ \mathrm{cm}^{-3}$, and the Coulomb logarithm is chosen to be $\Lambda_c = 16$. The initial energy of the test particles is set at 1.5 keV. The energy of the background particles is set at 2.25 keV for heating case, and 1.125 keV for cooling case.

Figs. 6 and 7 show the time history of the mean energy of the test species during heating and cooling with different grid resolutions. The density is conserved, and the momentum remains zero during the process. The mean energy evolution at different resolutions shows the convergence to the field particle energy. It can be seen the initial development follows the analytic curve very closely, because the test distribution is initialized as a

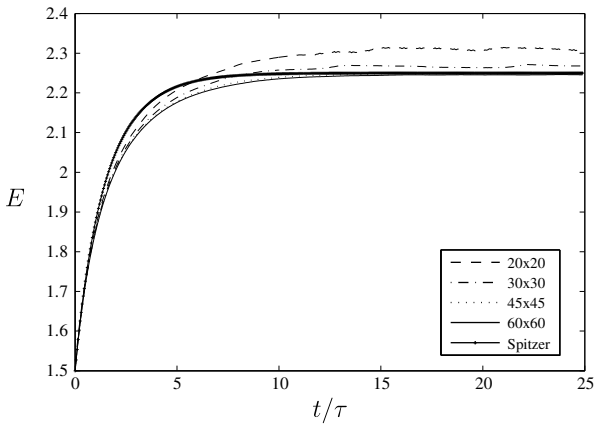


Fig. 6. Mean energy evolution of Spitzer thermalization. The field particle is Maxwellian with energy 2.25 keV and the test particle at $t = 0$ is Maxwellian with energy 1.5 keV.

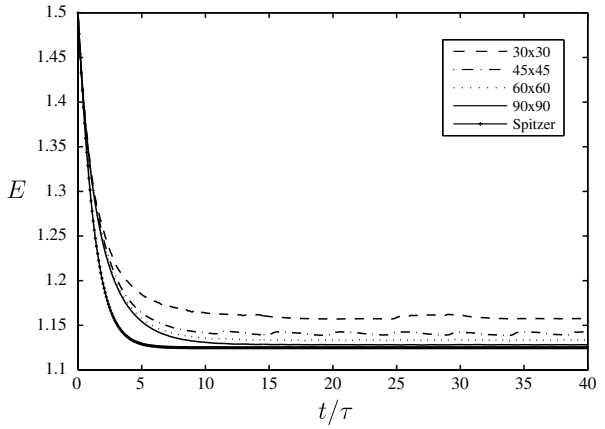


Fig. 7. Mean energy evolution of Spitzer thermalization. The field particle is Maxwellian with energy 1.125 keV and at the test particle at $t = 0$ is Maxwellian with energy 1.5 keV.

Maxwellian. As the collision evolves, the relaxation rate starts to deviate from the analytic curve. This is because particles with different energies have different collision rates, with higher energies relaxing more slowly, and as a result the distribution of the test particles is no longer a Maxwellian during the relaxation process [16]. Close to the thermal equilibrium, however, the numerical curve follows the theory again, because the distribution function has returned to a Maxwellian with a mean energy close to that of the background particles. The time history for the cooling case shows a similar progression.

### 5.3. Return to isotropy from loss cone distribution

In this section we examine the relaxation to Maxwellian due to collision effects for test particles with a initial loss cone in the velocity space. Loss cone situations are common in magnetized-plasma confinement experiments, e.g. the edge plasma of a toroidal magnetic-fusion device or in an open-ended magnetic-mirror device. In such cases, particles located in a certain part of the velocity space escape to material walls, leaving behind a velocity-space region with no particles inside. This region is often called a loss cone. In the presence of collisions, however, particles located in other parts of velocity space will be scattered into the loss cone. If we imagine confining such a plasma and preventing loss to material walls, the loss cone will gradually fill up. This collisional process is termed return-to-isotropy, and the distribution function for the test particles eventually recovers a Maxwellian form.

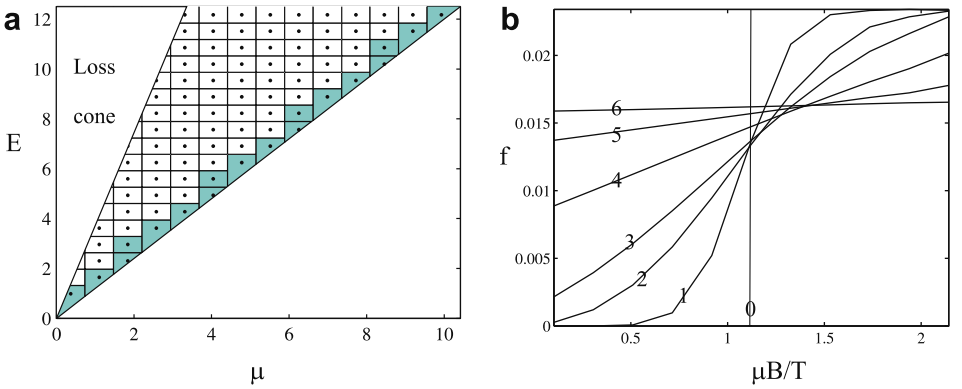


Fig. 8. (a) Loss cone in $(\mu, E)$ velocity space, (b) distribution function $f$ as a function of $\mu B/T$ for at a fixed energy $E/T = 2.387$ during the filling of the loss cone. The normalized times (by the basic relaxation time $\tau$) for the curves 0–6 are $t/\tau = 0, 0.027, 0.135, 0.27, 0.675, 1.35, 2.7$.

Fig. 8a shows a typical loss cone in velocity space for purely magnetic mirror trapping. In this case the loss cone boundary is a straight line in $(\mu, E)$ space whose slope is set by the maximum of the $B$ field. The parameters chosen for the loss cone problem are as follows: The slopes of the loss cone and the $v_\parallel = 0$ boundaries are $B_{\max} = 2.0$ and $B_{\min} = 1.2$. The initial condition for the distribution function $f$ is a Maxwellian at $T_\alpha = 1$ but with the loss cone region set to zero. The initial normalized density is thus 0.6991 instead of unity. Both the test and background particles have twice the proton mass and unit charge. The background temperature is $T_\beta = 1$. The grid resolution is $N_\mu = 50$ and $N_E = 45$, the number of mesh points in $(\mu, E)$, respectively. Let $T_0 = 1$ keV be the reference temperature, the magnetic moment is normalized by $T_0/B_{\min}$, and the normalized energy is $E/T_0$. Fig. 8b shows $f$ as a function of $\mu$ at a fixed $E/T_0 = 2.387$ at different times of the loss-cone filling. Time is normalized as before by the initial basic relaxation time $\tau$. Starting from a step function at time $t = 0$, the loss cone (left) is gradually filled by particles from the trapped region (right), so the increase of $f$ in the loss cone is accompanied by the decrease of $f$ in the trapped region. Eventually, the loss cone is completely filled, and $f$ is relaxed to a Maxwellian independent of $\mu$, as shown by the curve 6 in 8b.

## 6. Summary and discussion

In this paper, a new fourth-order finite-volume algorithm is developed for the Fokker–Planck collision operator (FPO) for highly magnetized plasmas. The velocity-space coordinates are chosen to be the parallel velocity $v_\parallel$ and magnetic moment $\mu$. Based on two-dimensional Gaussian quadrature, the finite-volume reconstruction scheme can be applied to arbitrary unstructured meshes in $(v_\parallel, \mu)$ coordinates, including those generated by directly mapping a mesh from another set of coordinates. The numerical FPO conserves the particle density to roundoff error, and the momentum and energy to fourth-order accuracy.

As an application, we compute the linearized collision operator in constants-of-motion coordinates $(\mu, E)$. A cell cutting and merging method is employed at the turning point boundary, and the resulting mesh is mapped into $(v_\parallel, \mu)$ coordinates analytically. A 15-cell stencil is devised for the reconstruction scheme, and the fluxes are evaluated using fourth-order Gaussian quadrature. The numerical accuracy of the reconstruction scheme is verified using a drifting Maxwellian. The FPO is used to compute the thermalization of a test distribution with a fixed Maxellian background, for both heating and cooling cases, and the return to isotropy of an initial distribution function with an empty loss cone. The numerical results agree well with theoretical predictions.

Note that for spatially non-homogeneous magnetic fields, although the present FPO *alone* is fourth-order accurate and number-conserving in velocity space, with the addition of convection in physical space via the spatial-orbit portion of the kinetic equation, it is non-trivial to maintain the same accuracy and conservation for the *whole* numerical algorithm. This difficulty is partially due to the fact that the size and shape of the cut-cells at the $v_\parallel = 0$ boundary change along the particle orbits, which presents a challenge for spatial convection schemes. In addition, the coupling in velocity space at different spatial points, due to parallel streaming and radial drift, makes the reconstruction scheme in velocity space susceptible to truncation errors in physical space. The most suitable choice of discretization schemes for convection, e.g. finite difference versus finite volume, and the best choice of velocity coordinates, to achieve overall optimal accuracy and conservation in coupling Vlasov solvers with the FPO, is a subject for future research.

The present method may be extended in several ways. Firstly, the reconstruction scheme is not specific to velocity space; it can be applied to configuration space as well. Secondly, though focused on the linear FPO in this paper, the algorithm is directly applicable to a fully nonlinear collision operator once the Trubnikov–Rosenbluth potentials are calculated. Lastly, the reconstruction scheme can be readily generalized to three dimensions.

## Acknowledgments

This work was performed under the auspices of the US Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48 at LLNL. The authors are grateful for stimulating discussions with Drs. B. Cohen, M. Dorr, J. Hittinger, G. Kerbel and W. Nevins.

Table 3
Abscissa and weights of fourth-order Gaussian quadrature on a standard triangle

| $i$ | $w_i$ | $\tilde{x}_i$ | $\tilde{y}_i$ |
|---|---|---|---|
| 0 | 0.144315607677787 | 0.333333333333333 | 0.333333333333333 |
| 1 | 0.095091634267285 | 0.081414823414554 | 0.459292588292723 |
| 2 | 0.095091634267285 | 0.459292588292723 | 0.081414823414554 |
| 3 | 0.095091634267285 | 0.459292588292723 | 0.459292588292723 |
| 4 | 0.103217370534718 | 0.658861384496480 | 0.170569307751760 |
| 5 | 0.103217370534718 | 0.170569307751760 | 0.658861384496480 |
| 6 | 0.103217370534718 | 0.170569307751760 | 0.658861384496480 |
| 7 | 0.032458497623198 | 0.898905543365938 | 0.050547228317031 |
| 8 | 0.032458497623198 | 0.050547228317031 | 0.898905543365938 |
| 9 | 0.032458497623198 | 0.050547228317031 | 0.050547228317031 |
| 10 | 0.027230314174435 | 0.008394777409958 | 0.263112829634638 |
| 11 | 0.027230314174435 | 0.008394777409958 | 0.728492392955404 |
| 12 | 0.027230314174435 | 0.263112829634638 | 0.008394777409958 |
| 13 | 0.027230314174435 | 0.263112829634638 | 0.728492392955404 |
| 14 | 0.027230314174435 | 0.728492392955404 | 0.008394777409958 |
| 15 | 0.027230314174435 | 0.728492392955404 | 0.263112829634638 |

## Appendix A. Gaussian quadrature

The fourth-order Gaussian quadrature on the standard triangle $\triangle = \{(0,0),(0,1),(1,0)\}$ is given by

$$\int_{\triangle} f(\tilde{x},\tilde{y})\,\mathrm{d}\tilde{x}\,\mathrm{d}\tilde{y} = \sum_{i=0}^{15} w_i f(\tilde{x}_i,\tilde{y}_i), \tag{46}$$

where the standard abscissa and weights are given in Table 3.

## References

[1] M.N. Rosenbluth, W.M. MacDonald, D.L. Judd, Fokker–Planck equation for an inverse-square force, Phys. Rev. 107 (1957) 1–6.
[2] J. Killeen, G.D. Kerbel, M.G. McCoy, A.A. Mirin, Computational Methods for Kinetic Models of Magnetically Confined Plasmas, Springer-Verlag, 1986.
[3] J.S. Chang, G. Cooper, A practical difference scheme for Fokker–Planck equations, J. Comput. Phys. 6 (1) (1970) 1–16.
[4] E.M. Epperlein, Implicit and conservative difference scheme for the Fokker–Planck equation, J. Comput. Phys. 112 (2) (1994) 291–297.
[5] I.K. Khabibrakhmanov, G.V. Khazanov, The spectral collocation method for the kinetic equation with the nonlinear two-dimensional coulomb collisional operator, J. Comput. Phys. 161 (2) (2000) 558–575.
[6] L. Chacon, D.C. Barnes, D.A. Knoll, G.H. Miley, An implicit energy-conservative 2D Fokker–Planck algorithm: I. Difference scheme, J. Comput. Phys. 157 (2) (2000) 618–653.
[7] C. Bueta, S. Cordierb, P. Degondc, M. Lemouc, Fast algorithms for numerical, conservative, and entropy approximations of the Fokker–Planck–Landau equation, J. Comput. Phys. 133 (2) (1997) 310–322.
[8] L. Pareschi, G. Russo, G. Toscani, Fast spectral methods for the Fokker–Planck–Landau collision operator, J. Comput. Phys. 165 (1) (2000) 216–236.
[9] F. Filbet, L. Parechi, A numerical method for the accurate solution of the Fokker–Planck–Landau equation in the nonhomogeneous case, J. Comput. Phys. 179 (1) (2002) 1–26.
[10] N. Crouseilles, F. Filbet, Numerical approximation of collisional plasmas by high order methods, J. Comput. Phys. 201 (2) (2004) 564–572.
[11] Y. Idomura, S. Tokuda, Y. Kishimoto, Global gyrokinetic simulation of ion temperature gradient driven turbulence in plasmas using a canonical Maxwellian distribution, Nucl. Fusion 43 (2003) 234–243.
[12] X. Xu, Z. Xiong, M. Dorr, J.A. Hittinger, J. Candy, K. Bodi, B.I. Cohen, R.H. Cohen, P. Colella, G. Kerbel, S. Krasheninnikov, W.M. Nevins, H. Qin, T.D. Rognlien, P.B. Snyder, M.V. Umansky, Edge gyrokinetic theory and continuum simulations, Nucl. Fusion 47 (2007) 809–816.
[13] B.A. Trubnikov, Particle interactions in a fully ionized plasma, Rev. Plasma Phys. 1 (1965) 105–204.
[14] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (2) (1999) 209–240.
[15] G. Kerbel, Z. Xiong, Numerical methods for nonlinear Fokker–Planck collision operator in TEMPEST, 48th APS Annual Meeting of the Division of Plasma Physics, October 30– November 3, vol. 51(7), Bullet. Amer. Phys. Soc., Philadelphia, PA, 2006, p. 158.
[16] K. Nanbu, Theory of cumulative small-angle collisions in plasmas, Phys. Rev. E 55 (4) (1997) 4642–4652.